

The opinion in support of the decision being entered today was *not* written for publication and is *not* binding precedent of the Board.

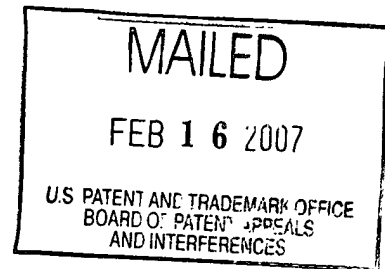
UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte EITAN FARCHI, THOMAS JOSEPH PAVELA, SHMUEL UR,
and AVI ZIV

Appeal No. 2006-2958
Application 09/990,802
Technology Center 2100

ON BRIEF



Before BARRY, HOMERE, and LUCAS, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants appeal from the Examiner's final rejection of claims 1 through 21 pursuant to 35 U.S.C. § 134. We have jurisdiction under 35 U.S.C. § 6(b) to decide this appeal.

The Examiner rejected claims 1 through 21 as follows:

- A. Claims 1, 6 through 8, 13 through 15, 20 and 21 stand rejected under 35 U.S.C. § 102 (b) as being anticipated by Chen .
- B. Claims 5, 12 and 19 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over Chen.
- C. Claims 2, 3, 9, 10, 16 and 17 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over the combination of Chen and Winder.
- D. Claims 4, 11 and 18 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over the combination of Chen and Reinhardt.

The Examiner relies on the following references:

Chen	5,673,387	Sep. 30, 1997
Reinhardt	5,778,169	Jul. 7, 1998

Winder, "Managing data through naming standards," IEEE Software, vol.7.4, 84-85, (1990).

Independent claim 1 is illustrative and representative of the Appellants' invention. It reads as follows:

1. A method using a computer system for collecting persistent code coverage data for a computer program, the computer program comprising program source code statements, the method comprising the steps of:

identifying the computer program for which the persistent code coverage data should be collected;

dividing the program source code statements of said computer program into a plurality of code coverage tasks;

generating a persistent unique name for each of the code coverage tasks of said plurality of code coverage tasks;

inserting coverage points into the computer program source code for each of the code coverage tasks to produce an instrumented program;

compiling and linking the instrumented program into a program executable;

identifying a set of test cases from a plurality of test cases to be run for the code coverage data collection purposes;

creating a code coverage database using the code coverage tasks and the identified set of test cases;

running the program executable with a test case from the identified set of test cases and writing the information about the test case and the coverage points that are executed into an output file, until all the test cases have been run; and

processing the information contained in the output file into code coverage data and populating the code coverage database with said code coverage data.

First, Appellants contend that Chen does not anticipate claims 1, 6 through 8, 13 through 15, 20 and 21. Particularly, Appellants contend that Chen does not fairly teach or suggest assigning a persistent unique name for each of a plurality of code coverage tasks. (Appeal Brief, page 10). Second, Appellants contend that claims 2 through 5, 9 through 12, and 16 through 19 would not have been obvious over Chen alone or in combination with

Winder or Reinhardt.¹ Particularly, Appellants contend that neither Winder nor Reinhardt cures the deficiencies of Chen. Appellants also contend that the Examiner failed to establish sufficient motivation to combine Chen with Winder. Therefore, Chen alone or in combination with Winder or Reinhardt would not have rendered claims 2 through 5, 9 through 12, 16 through 19 unpatentable. (Appeal Brief, pages 13 and 14; Reply Brief, pages 2 and 3).

The Examiner contends that Chen's naming of entities corresponds to Appellants' persistent unique naming of code coverage tasks, as claimed. (Examiner's Answer, page 4). The Examiner also contends that it would have been obvious to incorporate Winder's naming conventions into Chen's selective regression testing of software to yield the claimed invention (Examiner's Answer, page 14).

We affirm.

¹ This decision considers only those arguments that Appellants submitted in the Appeal and Reply Briefs. Arguments that Appellants could have made but chose not to make in the Briefs are deemed to have been waived. See 37 CFR 41.37(c) (1) (vii) (eff. Sept. 13, 2004). See also In re Watts, 354 F.3d 1362, 1368, 69 USPQ2d 1453, 1458 (Fed. Cir. 2004).

ISSUES

The *pivotal* issues on appeal before us are as follows:

- (1) Under 35 U.S.C. § 102 (b), does Chen's disclosure anticipate the claimed invention when Chen teaches a persistent unique name for each of a plurality of code coverage tasks?
- (2) Under 35 U.S.C. § 103 (a), would one of ordinary skill in the art at the time of the present invention, have found sufficient motivation to combine Chen and Winder by integrating naming conventions into attributes of selected entities?

FINDINGS OF FACT

Appellants invented a method, apparatus and article of manufacture for selective regression testing of a software suite. Particularly, the invention divides a program source code into a plurality of basic blocks of codes (code coverage tasks) and assigns a persistent unique name to each of the plurality code coverage tasks. (Spec. page 12). The created persistent unique names are then collected across software versions to identify which subsets of the test suite must be run in order to test a new version of the software. (Spec. page 13).

Chen discloses a regression testing method and system for selecting which test units of the test suite to be re-run in a new software version after a

previous version of the software was modified. (Col. 1, lines 10-12). As depicted in figure 1, Chen teaches partitioning the software source code into basic code entities, grouped as part of different test units. Chen further teaches that each entity has, inter alia, an entity kind, a name and a checksum. (Col. 9, lines 1-2). Chen also teaches that two entities match if they have the same name and entity kind. If the checksums of the two entities match, the source code texts of the two entities are probably identical. (Col. 11, lines 20-21, lines 46-49). These entity attributes are compared across software versions to determine which test units to re-run depending on which entities were previously modified. (Col. 2, lines 47-60, Col. 12, lines 9-18).

Winder teaches using naming conventions in software management. Particularly, Winder teaches selecting meaningful program variables and routine names in software development so they can be readily identified and reused (Page 84).

PRINCIPLES OF LAW

1. ANTICIPATION

It is axiomatic that anticipation of a claim under § 102 can be found only if the prior art reference discloses every element of the claim. See In re King, 801 F.2d 1324, 1326, 231 USPQ 136, 138 (Fed. Cir. 1986) and

Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 1458, 221 USPQ 481, 485 (Fed. Cir. 1984).

2. OBVIOUSNESS

On appeal, Appellants bears the burden of showing that the Examiner has not established a legally sufficient basis for combining the teachings of the references that the Examiner relied upon. Appellants may sustain this burden by showing that the Examiner has failed to provide sufficient evidence to support that one having ordinary skill in the art would have combined disclosures of the references, as proposed by the Examiner to yield Appellant's invention. United States v. Adams, 383 U.S. 39 (1966); In re Kahn, 441 F.3d 977, 987-988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006); DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick, Co., 464 F.3d 1356, 1360-1361, 80 USPQ2d 1641, 1645 (Fed. Cir. 2006). The mere fact that all the claimed elements or steps appear in the prior art is not *per se* sufficient to establish that it would have been obvious to combine those elements. United States v. Adams, *supra*, Smith Industries Medical Systems, Inc. v. Vital Signs, Inc., 183 F.3d 1347, 1356, 51 USPQ2d 1415, 1420 (Fed. Cir. 1999). However, "[a]s long as some motivation or suggestion to combine the references is provided by the prior art taken as a whole, the law does not require that the references be combined for the reasons

contemplated by the inventor.” In re Beattie, 974 F.2d 1309, 1312, 24 USPQ2d 1040, 1042 (Fed. Cir. 1992). Motivation to combine references under 35 U.S.C. § 103 must come from a teaching or suggestion within the prior art, within the nature of the problem to be solved, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources, to select particular elements, and to combine them as combined by the inventor. Ruiz v. A.B. Chance Co., 357 F.3d 1270 1276-77, 69 USPQ2d 1686 1691, (Fed. Cir. 2004).

“[A]n implicit motivation to combine exists not only when a suggestion may be gleaned from the prior art as a whole, but when the ‘improvement’ is technology-independent and the combination of references results in a product or process that is more desirable, for example because it is stronger, cheaper, cleaner, faster, lighter, smaller, more durable, or more efficient . . . In such situations, the proper question is whether the ordinary artisan possesses knowledge and skills rendering him *capable* of combining the prior art references.”

DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co., 464 F.3d 1356, 1368, 80 USPQ2d 1641, 1651 (Fed. Cir. 2006).

ANALYSIS

1. ANTICIPATION

The Examiner properly found that Chen anticipates the claimed invention. The Examiner properly recognized that Chen's entities or blocks of software codes must persist across software versions. Such a finding is reasonable since Chen's regression testing method only performs re-runs for test units with entities that were modified in a previous version of the software. Test units with unmodified entities are reused in subsequent versions of the software. Therefore, such unmodified entities or block of codes must persist across different versions of the software. Further, the Examiner properly found that Chen's entities are assigned unique names. This finding is also reasonable since Chen compares the name, checksum and kind attributes of each entity to determine whether two entities are the same, and that they have identical blocks of codes. After considering the entire record before us, we find that the Examiner did not err in rejecting claim 1 over Chen. We also find that the Examiner did not err in rejecting claims 6 through 8, 13 through 15, 20 and 21 over Chen.

2. OBVIOUSNESS

Next, the Examiner properly combined the teachings of Chen and Winder. The Examiner properly found that Winder's teaching of using name conventions in a software development environment would reinforce Chen's matching of entity attributes. One of ordinary skill in the art would have readily recognized that the suggested combination is proper to yield the claim limitation, which particularly deals with assigning unique names to blocks of software codes. After considering the entire record before us, we find that the Examiner did not err in rejecting claims 2, 3, 9, 10, 16 and 17 over the combination of Chen and Winder. We also find that the Examiner did not err in rejecting claims 4, 5, 11, 12, 18 and 19 over Chen, taken alone or in combination with Reinhardt.

CONCLUSION OF LAW

On the record before us, Chen's disclosure anticipates the claimed invention under 35 U.S.C. § 102 (b) as itemized above by teaching a persistent unique name for each of a plurality of code coverage tasks. Further, one of ordinary skill in the art, at the time of the present invention, would have found sufficient motivation to combine Chen and Winder under 35 U.S.C. § 103 (a) by integrating naming conventions into attributes of selected entities.

DECISION


No time period for taking any subsequent action in connection with
this appeal may be extended under 37 C.F.R. § 1.136(a) (1) (iv).

AFFIRMED



LANCE LEONARD BARRY)
Administrative Patent Judge)

)
)
) BOARD OF PATENT


JEAN R. HOMERE) APPEALS AND
Administrative Patent Judge) INTERFERENCES


JAY P. LUCAS)
Administrative Patent Judge)

JH/gw

Appeal No. 2006-2958
Application No. 09/990,802

IBM CORPORATION
C/O KENYON & KENYON LLP
333 W. SAN CARLOS STREET SUITE 600
SAN JOSE, CA 95110